



Effective Use of Local BLAST Databases

Using tools from blast+ to create customized databases for local search needs

<https://www.ncbi.nlm.nih.gov/books/NBK1762>

National Center for Biotechnology Information • National Library of Medicine • National Institutes of Health • Department of Health and Human Services

Introduction

NCBI makes the standalone blast+ package [1, 2, 3] freely available from the FTP site. Using this package, you can perform BLAST searches locally against the standard set of preformatted BLAST databases made available by NCBI [4]. More importantly, you will be able to search against your own custom sequence collections by making them into BLAST databases, by formatting them using the makeblastdb tool included in the blast+ package. Here we will address a few representative scenarios and demonstrate how this makeblastdb tool should be used.

Use Cases

The following use cases assume that the blast+ is configured with the path and BLASTDB environment variables properly specified. The working directory will be the BLASTDB specified directory, where database files are located.

Use Case 1: create a database alias to point to a taxonomic subset within a preformatted database from NCBI

Version 5 of the preformatted BLAST databases have taxonomic ID encoded for each entry, which can be used in search to limit a BLAST search through the “`-taxids <comma-separated list>`” switch to increase the blast search’s speed and clarity. The equivalent “`-taxidlist <taxid list file>`” switch is for large set of taxids, with the input being a plain text file, with taxids in one record per line format. Since this taxid encoding is at the species level for most entries, the input list can be difficult to identify and manage and database alias helps us get around that.

Step 1. Collect species level taxids for the desired organism group using `get_species_taxids.sh`

```
$ get_species_taxids.sh -t 40674 > mammals_species_taxids
```

This tool calls relevant tools from the EDirect package [5] to collect all the species level taxids under the input taxid (-t 40674) and redirect the output taxid list to the specified file (> mammals_species_taxids).

Step 2. Make an alias to the target database using `blastdbaliastool`

```
$ blastdbaliastool -db nt -taxidlist mammals_species_taxids -dbtype nucl \
 -title "test mammal subset" -out mammals_nt
```

The command sets the target database (-db nt), specifies the input taxids (-taxidlist mammals_species_taxids) and the database type (-dbtype nucl), assigns a title to the output alias (-title “test mammal subset”), and names the output (-out mammals_nt). This test commands outputs the following to the console, the number for your test will change due to database update:

```
Created nucleotide BLAST (alias) database mammals_nt with 10736843 sequences
```

The actual database alias will be the name specified by the “`-out`” option.

Step 3. Check the newly generated alias using `blastdbcmd`

```
$ blastdbcmd -db mammals_nt -info
```

This simply calls the alias database (-db mammals_nt) to get the summary (-info). It generates the following console output:

```
Database: test mammal subset
 10,736,843 sequences; 55,011,377,075 total bases
```

```
Date: Jan 3, 2021 9:56 AM      Longest sequence: 99,791,824 bases
```

```
BLASTDB Version: 5
```

```
Volumes:
  nt.nal
```

This maps to the nt.nal since the preformatted nt is a multi-volume database that was tied together through the “`*.nal`” file.

Step 4. Run searches against the alias database

```
$ blastn -db mammals_nt -query query_fasta_file_name -out output_file_name ...
```

Best Practice: Keep the master database, the taxidlist file, and the database alias files intact for this setup to work. Call the database with its base name without the file extension. Avoid accidental removal of database files by NOT using /db directory as the working directory.

Use Case 2: create an alias to point to a specific list of entries within a preformatted database from NCBI

Many investigators focus their research on a specific topic. Reflected in blast searches, this is the need to search against only sequences with specific characteristics. We can do this by collecting the accessions for those sequences of interest through the Protein database, and make that list into a database alias. The focus of the example below is a collection of bacterial gyrase b sequences.

Step 1. Collect the sequence accessions for bacterial gyrase b using EDirect

```
$ esearch -db protein -query 'txid2[orgn] AND (gyrb OR "gyrase b")' | \
efetch -format acc > bacterial_gyrb_acc
```

Invoke esearch to search protein database (-db protein) with specified terms ('txid2[orgn] AND (gyrb OR "gyrase b")'), then pass the output (|) to efetch to request accession list (-format acc), and redirect result to a file (> bacterial_gyrb_acc).

Step 2. Create a database alias with the accession list for use with the protein nr database

```
$ blastdbaliastool -db nr -seqidlist bacterial_gyrb_acc -out bacteria_gyrb_subset
```

The command invokes blastdbaliastool, specifies the master database (-db nr), the sequence id input (-seqidlist bacterial_gyrb_acc), and the output alias database name (-out bacteria_gyrb_subset). The command generates the following console output at the time of test, your run will likely be different due to database update:

```
Created protein BLAST (alias) database bacteria_gyrb_subset with 230370 sequences
```

Step 3. Check the newly generated alias using blastdbcmd

```
$ blastdbcmd -db bacteria_gyrb_subset -info
```

This returns the following summary on this alias database

```
Database: nr limited by bacterial_gyrb_acc
          230,370 sequences; 120,872,191 total residues
```

```
Date: Jan 7, 2021 12:42 AM      Longest sequence: 74,488 residues
```

```
BLASTDB Version: 5
```

```
Volumes:
```

```
nr.pal
```

Step 4. blastp search this alias database and check the result

```
$ blastp -task blastp-fast -db bacteria_gyrb_subset -query Q_aa -outfmt 7 \
-max_target_seqs 1500 -out blastp_vs_alias.tsv
```

Step 5. Spot check the result

```
$ head -10 blastp_vs_alias.tsv
# BLASTP 2.11.0+
# Query: AHG06698.1 gyrase B, partial [uncultured Streptomyces sp.]
# Database: bacteria_gyrb_subset
# Fields: query acc.ver, subject acc.ver, % identity, alignment length, mismatches, gap opens, q. start, q. end, s. start, s. end, eval, bit score
# 1500 hits found
AHG06698.1      WP_006134081.1  100.000 409      0      0      1      409      132      540      0.0      841
AHG06698.1      WP_031018111.1  100.000 409      0      0      1      409      140      548      0.0      840
AHG06698.1      AHG06698.1     100.000 409      0      0      1      409      1      409      0.0      839
AHG06698.1      WP_033273241.1  98.289   409      7      0      1      409      139      547      0.0      831
AHG06698.1      ALV50900.1    98.289   409      7      0      1      409      139      547      0.0      830
```

The following time comparison demonstrated significantly improved search time with the same search settings.

Searching against alias marked subset	Searching against whole nr
<pre>\$ time blastp -task blastp-fast -db bacteria_gyrb_subset -query Q_aa -outfmt 7 -max_target_seqs 1500 >/dev/null</pre> <pre>real 0m12.086s user 0m10.341s sys 0m1.561s</pre>	<pre>\$ time blastp -task blastp-fast -db nr -query Q_aa -outfmt 7 -max_target_seqs 1500 >/dev/null</pre> <pre>real 13m34.743s user 11m26.354s sys 1m59.546s</pre>

Use Case 3: create a database from a custom collection of FASTA sequences

Preformatted blast databases from NCBI cannot satisfy the research need for all the users out there. In addition, certain sequences from NCBI may not be included in those preformatted blast databases. To search against these collection of sequences, we will need to convert the FASTA sequences into a blast database first, using makeblastdb. This tool also has the capability to add taxid to input sequences if their sequence id and taxid mapping are provided. We will demonstrate this using a set of sequences from the NCBI Protein database, retrieved through tools found in EDirect package.

Step 1. Get the FASTA input sequences

```
$ esearch -db protein -query 'Gammaproteobacteria[orgn] AND (gyrb OR "gyrase b")' | \
efetch -format fasta > g_proteobac_gyrb.faa
```

The command is very similar to the one use in Case 2, with a narrower taxonomic scope. The requested output is FASTA (-format fasta) and the output is redirect to a file (> gamma_proteobac_gyrb.faa).

Step 2. Get the taxid mapping file

```
$ esearch -db protein -query 'Gammaproteobacteria[orgn] AND (gyrb OR "gyrase b")' | \
esummary | xtract -pattern DocumentSummary -element AccessionVersion TaxId > g_proteobac_gyrb_taxid.map
```

The command searches protein database (esearch -db protein) with specified query (-query '...'), passes the result to esummary (| esummary) to retrieve the summary XML. The output XML is then passed to xtract XML parser (| xtract) to check each record (-pattern DocumentSummary), extract out the accession.version and taxid (-element AccessionVersion TaxId). The resulting tab-delimited list of accession.version and their corresponding taxids are saved to the specified file (> gamma_proteobac_gyrb_taxid.map).

This taxid map file is simply a two-column text file, with accession in the first column and taxid in the second column:

```
QQJ85282.1      28144
QQJ90070.1      731
...
```

For custom FASTA file, makeblastdb uses the first string in the definition line (known as defline, marked by ">" sign) as seqid. For a custom FASTA sequence with 884 as its taxid:

```
>partial_ispA_seq Wolinella succinogenes partial IspA sequence
MNLLESVMEGFERFLEEAPLFEFGFPHYNEYLWEMVRNGGKRFRPRLLLGV/SALAPLLVKSAYAPALAL
EILHTYSLIHDLPAMDNAATRRGHPTLHVKYDEASAVLAGDALTNTAFYLLAQAPLGSDTKVALVREL
...
```

The taxid map file should look like this:

```
partial_psba_seq      884
```

Step 3. Make the protein FASTA sequences into a custom BLAST database with taxid mapping

```
$ makeblastdb -in g_proteobac_gyrb.faa -dbtype prot -parse_seqids -taxid_map g_proteobac_gyrb_taxid.map \
-title "gamma proteobacteria gyrb, with taxid mapping" -out g_proteob_gyrb
```

The command invokes makeblastdb to read an input file (-in g_proteobac_gyrb.faa) and make a protein database (-dbtype prot). It enables seqid parsing (-parse_seqids) and taxid mapping (-taxid_map g_proteobac_gyrb_taxid.map). It adds a title in double quotes for clarity (-title "gamma proteobacteria gyrb, with taxid mapping"), and names the output (-out g_proteob_gyrb).

The command outputs the following to the console:

```
Building a new DB, current time: 01/11/2021 08:59:21
New DB name: /export/home/tao/g_proteob_gyrb
New DB title: gamma proteobacteria gyrb, with taxid mapping
Sequence type: Protein
Keep MBits: T
Maximum file size: 1000000000B
```

```
Adding sequences from FASTA; added 157194 sequences in 5.54401 seconds.
```

Step 4. Check the resulted database

```
$ blastdbcmd -db g_proteob_gyrb -entry all -outfmt "%a %T %S" | head -3
```

Use blastdbcmd to check this newly created database (-db g_proteob_gyrb), asking for complete dump (-entry all) in specified accession, taxid, and scientific name format (-outfmt "%a %T %S"). For brevity, only the first 3 records is shown through shell's head (| head -3). The command outputs the following to the console:

```
QQL53508.1 595 Salmonella enterica subsp. enterica serovar Infantis
QQL42496.1 263 Francisella tularensis
QQL39107.1 573 Klebsiella pneumoniae
```

Use Case 4: make an alias for the database created in Case 3

Sequence records in the NCBI Protein database have various sequence lengths. For some analysis, it may be preferable to search against those within certain length range. However, web protein blast and standalone blast+ do not allow this type of search limit directly.

The workaround is to take advantage of Entrez Protein to generate a list of sequence ids fit our desired criteria, generate an alias database with that list, then search against the alias database to achieve our blast search need. In this case, we will use the database generated in Case 3, as the parent, with the full-length gyrb as the desired target subset.

Step 1. Generate the sequence ids for full-length gyrb for gammaproteobacterial using EDirect

```
$ esearch -db protein -query 'gammaproteobacteria[orgn] ("gyrase b" OR gyrb) AND 750:850[slen]' | \
esummary | xtract -pattern DocumentSummary -element AccessionVersion > g_proteobac_gyrb_full_length.acc
```

Step 2. Generate the alias database with this sequence id list

```
$ blastdbAliastool -db g_proteob_gyrb -dbtype prot -seqidlist g_proteobac_gyrb_full_length.acc \
-title "gamma proeobac gyrb full length subset" -out gyrb_full_length_subset
```

This output the following to the console:

```
Created protein BLAST (alias) database gyrb_full_length_subset with 66773 sequences
```

Step 3. Verify the creation using blastdbcmd

```
$ blastdbcmd -db gyrb_full_length_subset -info
Database: gamma proeobac gyrb full length subset
66,773 sequences; 53,781,262 total residues
```

```
Date: Jan 11, 2021 9:57 AM Longest sequence: 2,906 residues
```

```
BLASTDB Version: 5
```

```
Volumes:
```

```
g_proteob_gyrb
```

Step 4. blastp search this subset limiting to *Salmonella enterica* (taxid: 28901)

```
$ blastp -query gyrb_mutant_fragment.faa -db gyrb_full_length_subset -taxids 28901 -outfmt "7 delim=; sacc
staxid ssciname bitscore evalue" -max_target_seqs 5
```

This invokes blastp program to use gyrb_mutant_fragment.faa as query (-query gyrb_mutant_fragment.faa [6]) and the alias as the database (-db gyrb_full_length_subset), limit the search to *Salmonella enterica* (-taxid 28901), and ask for custom tabular output (-outfmt "7 delim=; sacc staxid ssciname bitscore evalue"). Since we do not specify the output file, blastp prints the results to the console. The output contains the custom columns we specified (for demonstration purpose), separated by the delimiter we specified (delim=;). The ssciname field requires the presence of taxdb files provided by NCBI (<ftp.ncbi.nlm.nih.gov/blast/db/taxdb.tar.gz>), that is also packaged with all preformatted blast databases [4].

```
# Query: GyrB_fragment Salmonella enterica gyrb with two mutations
# Database: gyrb_full_length_subset
# Fields: subject acc., subject tax id, subject sci name, subject title
# 20 hits found
WP_024156120;28901;Salmonella enterica;DNA topoisomerase (ATP-hydrolyzing) subunit B [Salmonella enterica]
KTM83384;28901;Salmonella enterica;DNA gyrase subunit B [Salmonella enterica]
...
```

References

- Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, Madden TL. BLAST+: architecture and applications. BMC Bioinformatics. 2009 Dec 15;10:421. doi: 10.1186/1471-2105-10-421. PMID: [20003500](#); PMCID: [PMC2803857](#).
- Tao T. Standalone BLAST Setup for Unix. <https://www.ncbi.nlm.nih.gov/books/NBK52640/>
- Tao T. Standalone BLAST Setup for Windows PC. <https://www.ncbi.nlm.nih.gov/books/NBK52637/>
- Tao T, Madden T, and Camacho C. BLAST FTP Site. <https://www.ncbi.nlm.nih.gov/books/NBK62345/>
- Kans J. Entrez Direct: E-utilities on the Unix Command Line. <https://www.ncbi.nlm.nih.gov/books/NBK179288>
- Eaves DJ, Randall L, Gray DT, Buckley A, Woodward MJ, White AP, Piddock LJ. Prevalence of mutations within the quinolone resistance-determining region of gyrA, gyrB, parC, and parE and association with antibiotic resistance in quinolone-resistant *Salmonella enterica*. Antimicrob Agents Chemother. 2004 Oct;48(10):4012-5. doi: 10.1128/AAC.48.10.4012-4015.2004. PMID: [15388468](#); PMCID: [PMC521866](#).